Atty. Docket No. MS303531.2

BOUNDED-DEFERRAL POLICIES FOR GUIDING THE TIMING OF ALERTING, INTERACTION AND COMMUNICATIONS USING LOCAL SENSORY INFORMATION

by

Eric J. Horvitz

MAIL CERTIFICATION

I hereby certify that the attached patent application (along with any other paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on this date June 30, 2003, in an envelope as "Express Mail Post Office to Addressee" Mailing Label Number EV330021095US addressed to: Mail Stop: Patent Applications, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450

Himanshu S. Amin

Title: BOUNDED-DEFERRAL POLICIES FOR GUIDING THE TIMING OF ALERTING, INTERACTION AND COMMUNICATIONS USING LOCAL SENSORY INFORMATION

REFERENCE TO RELATED APPLICATIONS

This application is a continuation in part of U.S. Patent Application Serial No. 10/220,550 filed on August 30, 2002, entitled PRIORITIES GENERATION AND MANAGEMENT, which claims priority to PCT Application Serial No. PCT/US01/08710, filed on March 16, 2001, which claims the benefit of U.S. Provisional Patent Application Serial No. 60/189,801, filed on March 16, 2000, entitled ATTENTIONAL SYSTEMS AND INTERFACES.

This application is also a continuation in part of U.S. Patent Application Serial No. 10/220,419 filed on August 30, 2002, entitled NOTIFICATION PLATFORM ARCHITECTURE, which claims priority to PCT Application Serial No. PCT/US01/08711, filed on March 16, 2001, which claims the benefit of U.S. Provisional Patent Application Serial No. 60/189,801, filed on March 16, 2000, entitled ATTENTIONAL SYSTEMS AND INTERFACES. All of the aforementioned applications are incorporated herein by reference.

TECHNICAL FIELD

The present invention relates generally to systems and methods that facilitate communications between devices, systems, processes, and/or individuals. More particularly, the present invention relates to smart sensing of endpoint devices that are employed with various communications systems and in accordance with bounded deferral policies for minimizing the disruptiveness of notifications.

BACKGROUND OF THE INVENTION

Despite the impressive abilities of people to sense, remember, and reason about the world, cognitive abilities are extremely limited in well-characterized ways. In particular, psychologists have found that people wrestle with scarce attentional resources and limited working memory. Such limitations become salient when people are challenged with remembering more than a handful of new ideas or items in the short term, recognizing important targets against a background pattern of items, or interleaving multiple tasks.

These results indicate that people must typically inspect the world through a limited spotlight of attention. As such, most people often generate clues implicitly and explicitly about what they are selectively attending to and how deeply they are focusing. Findings about limited attentional resources have significant implications for how computational systems and interfaces are designed.

SUMMARY OF THE INVENTION

The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is not intended to identify key/critical elements of the invention or to delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

The present invention relates to a system and method to facilitate communications of important messages or communications. Policies are described that consider a user's current situation, including the consideration of an inferred cost of interruption, or directly sensed surrogates for such a cost, in decisions about the best time, within a deadline for delivering messages. Deadlines for delivery depend on the urgency of the information that is inferred or detected from the message sender, type, and content. If a suitable time is not detected within a deadline, the information is delivered at the deadline. If it is determined that a suitable time will not achieved within a deadline, the information is transmitted immediately. Suitable times for delivery can be determined

via the use of one or more sensors on or near endpoint devices, including accelerometers, microphones, touch sensing, and gaze and head pose detection. Other information, including appointment status as indicated on a user's calendar, time of day, and previously assessed patterns of availability can be employed in decisions about the deferral of alerts. Endpoint sensors, calendar information, and patterns of availability are also used to identify the likelihood that information will be received at a device. Such information can be passed back directly or in summary to a central notification manager or used locally in decisions about the salience and repetition of alerting.

One particular aspect of the present invention relates to systems and methods that facilitate efficient and timely communications between parties by mitigating disruptiveness associated with notifications. One particular aspect of the invention relates to employment of small devices (*e.g.*, telephones, PDAs, smart pens, watches, eyewear) in connection with message notification and/or best mode to effect communications. In accordance with the subject invention, small devices can be made aware or at least partly aware of various metrics relating to attentional status and/or location of users. Information determined and/or inferred by the small device(s) in connection with the attentional status and/or location can be shared between small devices as well as with disparate devices or systems (*e.g.*, a central Notification Managing system). The information can be disseminated individually, in serial or parallel *vis a vis* other devices, as well as aggregated. The information can be employed to facilitate providing a notification service and/or determining or inferring a best mode in which to effect communications with and between users.

One example aspect provides for employment of bounded-deferral policies wherein a local device commits to relaying a message that it has received before a message-specific deadline is reached; the device in accordance with the invention attempts to determine or infer a most appropriate time for interruption within an allotted period. Such determination or inference can employ statistical-based and/or probabilistic-based and/or utility-based (*e.g.*, benefit of interruption given cost of interruption) techniques. Devices in accordance with the subject invention can employ various sensing modalities (*e.g.*, MEMS-based sensors, temperature sensors, accelerometers, gyroscopes, light-based sensors, time-based sensors, GPS, 802.11 signal

strength, infrared proximity detectors, touch sensors, ...) in connection with learning or inferring an attentional status and/or location of users. With respect to sharing and/or sending sensed states, it is to be appreciated that all sensed states, subsets or summaries thereof can be communicated.

Another aspect of the invention provides for taking into consideration states of the device(s) and surrounding environment as well. For example, transmission reliability (transrel) of the device can be considered (*e.g.*, on a sliding scale representing the P(transrel | context), that is the likelihood of getting through on a device given context (is a function such as for example f(context) or f(sensed states).

To the accomplishment of the foregoing and related ends, certain illustrative aspects of the invention are described herein in connection with the following description and the annexed drawings. These aspects are indicative of various ways in which the invention may be practiced, all of which are intended to be covered by the present invention. Other advantages and novel features of the invention may become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic block diagram illustrating endpoint device communications in accordance with an aspect of the present invention.

Fig. 2 is a diagram an exemplary bounded deferral policy in accordance with an aspect of the present invention.

Fig. 3 is a diagram illustrating exemplary bounded deferral parameters in accordance with an aspect of the present invention.

Fig. 4 is a diagram illustrating application models for endpoint devices in accordance with an aspect of the present invention.

Fig. 5 illustrates an example endpoint device in accordance with an aspect of the present invention.

Figs. 6 and 7 are flow diagrams illustrating bounded deferral processing in accordance with an aspect of the present invention.

Fig. 8 is an influence diagram illustrating a model of attentional focus and workload in accordance with an aspect of the present invention.

Figs. 9 is an influence diagram illustrating a temporal Bayesian attentional model in accordance with an aspect of the present invention.

Fig. 10 is a conceptual overview of a notification platform in accordance with an aspect of the present invention.

Fig. 11 is a schematic diagram illustrating notification platform in accordance with an aspect of the present invention.

Fig. 12 is a diagram illustrating notification manager analysis in accordance with an aspect of the present invention.

Fig. 13 is a diagram illustrating real time reasoning in a notification platform in accordance with an aspect of the present invention.

Fig. 14 is a schematic block diagram of a prioritization system in accordance with an aspect of the present invention.

Fig. 15 is a schematic block diagram illustrating systematic cooperation between a notification engine and a context analyzer according to an aspect of the present invention.

Fig. 16 is a schematic block diagram illustrating a suitable operating environment in accordance with an aspect of the present invention.

Fig. 17 is a schematic block diagram of a sample-computing environment with which the present invention can interact.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a system and method to facilitate communications between parties, based on a consideration of context of a user, where context is taken to include such background information as a user's appointment status encoded on a user's digital calendar and information that is sensed locally through sensors on or near a device. We introduce two constructs and messaging policies that take advantage of such local sensing for alerting users about important information: *bounded deferral* and *transmission reliability*. *Bounded deferral* addresses the timeliness of alerts, providing a means for allowing a system to find a good time, while bounded the costs associated with the loss of value with delay of information. *Transmission reliability* informs a

notification system about the best device and alerting means available for communication a message.

Bounded-deferral policies define a deadline for making a user aware of a message containing information of value to the user, where the tolerance or deferral period is dependent on a determined time-dependent urgency for the information. Deadlines for alerting the user, or *deferral tolerances*, are assigned to messages that are transmitted to endpoint devices such as a mobile phone or handheld computer. Such tolerances may be assigned locally, at an endpoint device or by a more central notification manager. The deferral tolerances are considered by the endpoint device, in conjunction with stored or sensed information related to a user's context and availability. *Transmission reliability* refers to the probability that a message will get through to a user given the situation sensed by endpoint sensors and/or other relevant information such as information about a user's appointments as stored in a calendar.

Bounded deferral policies are employed, for example, wherein a local device commits to relaying a message that the device has received before a message-specific deadline is reached. The device then autonomously determines a suitable time for interruption within an allotted period. Endpoint devices can leverage multiple perceptual sensors including GPS, 802.11 signal strength, accelerometers, infrared proximity detectors, and touch sensors, for example.

Bounded deferral and transmission reliability can interact. Bounded deferral policies can leverage considerations of transmission reliability. For example, a user may be reachable with a loud alert but prefers to be notified with a gentle buzz, *e.g.,* via the vibration of a cell phone. If the transmission reliability of a gentle buzz becomes high enough before a deferral tolerance is reached, the user can be buzzed gently, bypassing the need for the loud alert.

In another aspect, bounded deferral can be applied to putting a caller on hold, (*e.g.,* while the user typing, etc.) and trying to break through to the user at a suitable time over some short time horizon (*e.g.,* providing a caller with an intermittent message such as "still trying..please continue to wait now, or press the # key to go directly to voice mail"). The system may be successful at finding a suitable context to put the call through or not, and thus give up and provide the user with other options such as rescheduling the

call and or forwarding the user to voice mail with an apology for the wait. This type of switching and decision-making can be executed by the endpoint device itself, or by a larger phone system, (*e.g.,* a corporate PBX, etc.) that has access to the local sensors. Also, bounded deferral policies can be can be coordinated with other parameters such as information about the user's location and/or context (*e.g.,* if I am in a movie theater, do not interrupt me unless an emergency message comes from my home after a deferral tolerance period has elapsed).

As used in this application, the terms "component," "service," "model," and "system" are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

As used herein, the term "inference" refers generally to the process of reasoning about or inferring states of the system, environment, and/or user from a set of observations as captured via events and/or data. Inference can be employed to identify a specific context or action, or can generate a probability distribution over states, for example. The inference can be probabilistic – that is, the computation of a probability distribution over states of interest based on a consideration of data and events. Inference can also refer to deterministic or logical reasoning techniques, including methods employed for composing higher-level events from a set of events and/or data. Such inference results in the construction of new events or actions from a set of observed events and/or stored event data, whether or not the events are correlated in close temporal proximity, and whether the events and data come from one or several event and data sources.

Referring initially to Fig. 1, a system 100 illustrates endpoint device communications in accordance with an aspect of the present invention. One or more

endpoint devices 110 are associated with one or more sensors 114 (*e.g.*, motion, proximity) and bounded deferral data 120 in order to facilitate communications with a user. The bounded deferral data 120 is typically determined and generated by a notification manager 130 that routes notifications from various notification sources 134 to the endpoint devices 110. It is to be appreciated that the endpoint devices 110 may also determine bounded deferral data 120 in accordance with the present invention.

Bounded deferral data 120 for the endpoint devices 110 is determined in consideration of a tolerated period, called a deferral period that is a generally a function of a notification sender and/or a type of message delivered. When messages are received by the endpoint device 110, the device employs its sensors 114 locally to determine a suitable time within an indicated bounded deferral period to alert a user. Typically, the more urgent messages are, the shorter the deferral period. For example, the notification manager 130 may have just received a message from a notification source 124 and attempts to deliver the message in accordance with a bounded deferral period specified by the bounded deferral data 120. Typically, the notification manager 130 makes general decisions about notification routing and relies on the endpoint device 110 to actually deliver the message to the user within the determined deferral period. However, the endpoint device 110 may detect that a user is currently involved in strenuous activity (*e.g.*, accelerometer indicating fast movement). As such, even though a message delivery deadline is approaching as defined by the bounded deferral data 120, the endpoint device 110 may still delay delivery of the message based upon detected activities or attentional state of the user.

It is noted that bounded deferral generally relates to the concept that messages are assigned locally or centrally with a bounded deferral tolerance that dictates a deadline for making a user aware of a message containing information of value to the user, where the tolerance or deferral is dependent on the urgency of the information. Also, a transmission reliability may also be considered, based potentially on locally sensed information which is related to a probability that a message will get through to a user given endpoint sensing and/or estimates given background information as will be described in more detail below.

Bounded deferral parameters can be determined *via* local sensors, calendar information, an alerting type, and/or time of day, for example to determine that a user is

too busy to receive an alert either now or for the next x minutes. If the deadline is reached and the alert has not yet been delivered, it is delivered at deadline. If a deadline will pass definitively (*e.g.,* as determined from calendar information) and there is no sense in waiting for a "better time," then the alert is passed immediately as there is nothing to be gained by waiting.

Sensors can be employed for various determinations such as determining when a user is currently busy and when a user is available to receive messages. Sensors can also indicate us what the transmission reliability is. Such sensor information can be passed directly off to a central notification manager, *e.g.,* a general notification platform that is deliberating about where to send messages, or in another application, an endpoint device itself can compute a transmission reliability (abbreviated as *transrel)* from its sensors and pass back the summary transmission reliability to a central notification manager which considers this in its deliberation. For example, a temperature sensor on a cell phone can indicate to the cell phone that it is indeed in a user's pocket right now, thus there is a high transrel to vibrating or ringing the cell phone to get through to the user.

In a conversational dialog system or aspect to the present invention, concepts of bounded deferral can be used to allow a system that performs interactive dialog, either to initiate a conversation or to continue a conversation that has been interrupted by the user's attention being diverted elsewhere for a task or another conversation. For example, in the case of an automated system that is working with a user on a task, whereby a dialog has been broken by someone stepping into a doorway and talking with the user, the system can wait to continue a certain amount of time before, apologizing and breaking in to continue, depending on the time-criticality of continuing the dialog.

It is noted that bounded deferral can be linked to many aspects of a user's experience. For example, bounded deferral policies can be linked to calendar information, where appointments are taken as deterministic bounds on a user's availability (*e.g.,* "user's appointment is ending in 10 minutes; thus it is okay to wait," versus, "user's appointment is ending in 20 minutes; the end of the appointment or task comes after the deferral tolerance. Thus, it is best to simply break in and alert the user immediately). Beyond calendar information, computing systems may be able to access information about the duration of other tasks. For example, tasks of predetermined

length may be available in contexts where a user is reviewing media (*e.g.,* a movie or play being watched by a user will end in 7.5 minutes, a commercial break will come at 14 minutes, etc.).

Similarly user's can specify various options about their availability based on the time of day and day of week (*e.g.,* a user may specify in preferences encoded in a notification manager, "Don't relay an alert to me before 8am and after 11pm unless you reach a deadline per the deferral tolerance associated with an alert."). A system can thus be charged with attempting to wait, and to hold off on alerting if possible, until the preferred period of time (e.g., between 8am to 11pm), but if it cannot wait, it will break through at other times. Also, bounded deferral policies can be dependent on the type of alerting (*e.g.,* device may vibrate gently immediately but if there is no confirmation before the deferral tolerance has been reached, the device alerts the user with a loud tone).

In another approach, rather than breaking through when a deferral tolerance has been reached, the endpoint device can be instructed to send a message back to a central notification manager or the sender of the alert, informing the notification manager that the device was unsuccessful at relaying the message.

It is noted that the notification manager 130 and endpoint devices 110 may employ decision-theoretic approaches when guiding or interrupting notifications to users and are described in more detail below. Global bounded deferral policies can be viewed as approximation of more detailed decision-theoretic analyses. In detailed decision-theoretic analyses, alerts are handled on a case-by-case basis, considering the detailed costs and benefits of alerting for each message and context. Bounded deferral policies allow for the specification of bounds on the total delay, and thus, total loss in the value of information with time for messages of different urgencies. When considering multi-message interactions, such as when a message breaks through to the user, other parties can be allowed to come through as well, even if they would not have broken through to the user on their own. In one example, a value for multiple messages may be determined that leads to a shorter deferral, *e.g.,* the sum of the value (or other function) of the value of independent messages.

Also, a system that has been holding back on several messages that have relatively long deferral tolerances (as they have low time criticality) that have not yet been met, may share the less critical messages at the time that a more time critical message, with a shorter deferral tolerance breaks through to a user. Such breakthroughs may incur most of the cost of information sharing, allowing other messages to pass through at low incremental cost. For example, consider the case where an alert about a meeting reminder to a person breaks through with enough time for the user to travel to a scheduled meeting, based on the user's current location and the location of the meeting. At the time of the breakthrough, the cost of receiving additional messages may not be significantly greater than the initial breakthrough cost. Thus a system, may alert the user with a message, "You have a meeting at the Doubletree Hotel in Bellevue in 20 minutes," and then after this alert is rendered and processed, share with the user messages of lower time criticality, "While I have your attention, tomorrow is Steven's birthday, and Joe Jones will be coming to town next week."

In order to route notifications to the user, the notification manger 130 and/or endpoint devices 110 can include one or more models for reasoning about user states (*e.g.,* attentional state, busyness). Such models can include substantially any type of system such as statistical/mathematical models and processes that include the use of Bayesian learning, which can generate Bayesian dependency models, such as Bayesian networks, naïve Bayesian classifiers, and/or Support Vector Machines (SVMs), for example. Other type models or systems can include neural networks and Hidden Markov Models, for example. Although elaborate reasoning models can be employed in accordance with the present invention, it is to be appreciated that other approaches can also utilized. For example, rather than a more thorough probabilistic approach, deterministic assumptions can also be employed (*e.g.,* no cell phone activity for *X* amount of time may imply by rule that user is not available by phone). Thus, in addition to reasoning under uncertainty as is described in more detail below, logical decisions can also be made regarding the status, location, context, focus, and so forth of users and/or associated devices.

Referring now to Fig. 2, a diagram 200 illustrates an exemplary bounded deferral policy in accordance with an aspect of the present invention. According to this aspect,

notifications or messages are not typically delivered until an available free state is reached unless a time bound is detected. For example, free states are illustrated at references 202 and 204. During busy states of the user (depicted as opposite to the free states 202, 204) a high and low priority message 206 and 208 are queued by a notification agent or manager (not shown). At 210, a time bound that was set as a max deferral time is reached for the high priority message and thus the high priority message is delivered to the user at 212. The low priority message 208 does not reach a time bound in the illustrated example of Fig. 2. Thus, the low priority message is not delivered until the next available free state at 204. In this manner, disruptiveness of notifications received by the user is mitigated. It is noted, that the time bounds can be influenced by the users context such as workload, number of messages received, and the time dependency of the notification content.

In accordance with the present invention, various algorithms and/or processes are provided for desktop and endpoint device alerting. These processes can be applied to multiple situations such as: (1) User present at desktop or endpoint device; (2) User away from desktop or endpoint device; and (3) User just returning or logging in to a desktop or endpoint device after being away.

For the case where a user is detected to be at a desktop or endpoint device, the following process can generally be applied:

1. When a notification is received, its age is set to zero and its priority is noted and a list of exceptions is checked.
2. If a "likely available" state is observed via monitoring the user's activities before the max deferral time for that urgency, the notification is passed through to the user.
3. Else, the notification is relayed when the deferral tolerance is reached for the notification as depicted above in relation to Fig. 2.

On average, because of the typical smatter of "likely available" states during typical desktop or endpoint device activities, most notifications will tend to be delivered before the max deferral times. However, user's will be more pleased on average with the notification system as notifications will tend more so to occur when the user is free than they would have been had notifications simply been passed through when notifications

are received. The probability that a free state will be reached generally increases with time—as there are more opportunities for detecting a likely available state with increasing amounts of time. As the probability of a likely free state increases with increasing amounts of times, lower priority messages will tend to occur with higher-likelihood during these likely free states, and the probability of being disrupted will grow with the increasing priority of the messages.

According to another aspect of the present invention, a display of notifications (*e.g.*, journal, browser, in-box, cell phone message box) can include multiple, or pooled notifications that have been waiting, so as to send to the user a single notification that contains chunks of grouped notifications. Such chunking can present the chunks of notifications in lists ordered by max priority, max age, or max priority by group, *etc.* For example, if a likely free state has not been detected, and that max deferral time has been reached by a high priority notification, and at the time the max deferral has been reached for the high priority notification, information can be included about the lower priority notifications that are pending in a grouped notification---even though the lower priority notifications will not have obtained an associated max deferral at this time. Several aspects are possible for this kind of chunking, including sending the main alert in a standard notification display, and summarizing other pending alerts in a list at the bottom of the display. Respective items can be clicked on and be reviewed and/or cleared by the user.

According to another aspect of the present invention, a calendar can be examined to enable users to specify uninterruptible meetings (*e.g.*, presentations, video conference, phone meetings) that should not be interrupted (*e.g.*, until some safe time, 10 minutes after end of meeting) except for notifications that are marked as immediate pass through. This can be generalized to utilizing a separate max deferral table and/or function for important meetings. This can be further generalized by enabling calendar items to be one of several classes of appointment and employ different max deferral tables or functions for different classes of meeting.

In another aspect of the present invention, instead of providing a few categories of priority, a continuous range can be provided, such as, 0-100 for an urgency score and the max deferral can be a function of the priority of the notification, including a variety of

linear and nonlinear functions (*e.g.,* exponential decay of max deferral time with increasing priority). For example:

$$\textit{max deferral(priority)} = e^{-k(priority)} \times 15 \text{ minutes}$$

which is equivalent to

$$\textit{max deferral(priority)} = e^{-k(priority)} \times \text{max deferral(0 priority)}$$

Additionally, users can specify contexts as a function of type of day (*e.g.,* weekend, holiday, weekday), time of day, and other basic contexts that change value assignments for different classes and subclasses of message (*e.g.,* e-mail, instant messenger communications from family versus business associates).

In another respect, a Notification Journal for items that have not yet been observed by the user can be provided. This can include maintaining a global Notification Journal for substantially all notifications—enabling users to return and access notifications that have been previously received, for example. This can also include providing for rich display and interaction. For example, a click on a journaled item in a Notification or endpoint device Window can bring up the notification. A click on the notification brings up more information or the appropriate user interface for the source of the notification. For example, clicking on a notification about an upcoming appointment brings up a full view of an appointment being referred to by the notification. Also, highlighted links can be displayed within notifications and enable users to jump to web pages, applications, or information associated with the notification. Furthermore, advertisements, special backgrounds and/or other branding information (from the source) can be displayed in the notification window, when a notification is rendered.

In another aspect, notifications with active durations, and/or with expiration dates, can be removed from an active queue after the date has passed. Notifications in a journal can be listed as expired if users are interested in seeing the history of this kind of activity. In addition, classes of notification can be tagged as being intrinsically replaceable by any update of information as identified by a Globally Unique Identifier (GUID), for example, in order to provide an update on the world state of information that the notification is reporting.

User Interface tools can be provided that enables users to append priority information to messages, or, more simply to do a normal Send or a *When Free* send. A

When Free send would be ported through the bounded deferral system described above; a normal send can act as a non-bounded communication. Notifications can also be tagged with application-specific (or life-specific) *contexts* from a set of contexts (*e.g.*, MS Word at focus, MS Outlook at focus, *etc.*) and render the notifications within the active context if it has not expired. For example, an assistance tip about a word processor usage rendered *via* a notification system should generally be provided when the word processor is at focus. If the application is not at focus, the tip should simply be journaled.

More advanced features can also be provided. For example, a frequency of “likely available times” for a user can be observed and learned, when users are working at a desktop or using an endpoint device, and the frequency with which alerts are received by the user in each class, and infer the expected time until the next likely free state, from a user's activity (based on application, time of day, expected user location, *etc.*). This information can be employed to automatically set the max deferral times for a respective notification priority class so as to enable the notification system to bound the probability of being disturbed for each priority class of alerts. This can be set by default, or can enable users to specify a probability for each priority class, and thus, inform the system that they do not want to be disturbed (that is, alerted when busy) for more than say, 5% of the time for low priority alerts and more than 10% of the time normal priority alerts, and 25% of the time for high priority alerts, *etc.* That is, users can specify a target "tolerated probability" of disruption for a respective priority class and the system can set the max deferral times for the classes.

Confirmation can be received that important notifications have been observed, for example, a convention can be employed that hovering over a notification is a signal that “I got it,” and utilize this feedback as an option that a user can turn in *via* a profile. That is, users can opt to turn on the option:

[ ] Continue to notify me about critical information every [x] minutes until I confirm with a mouse over or keystroke on an endpoint device.

When a user has been away from a desktop device for more than x minutes (set as default or by user specified amount of time), desktop events can be deferred, and instead notifications can be sent to a mobile or endpoint device. Similar max deferral times can be employed as specified for desktop alerting, or instead access an alternate set of max

deferral times for the "away" condition. That is, another table or function for controlling the max deferral time for the *away* situation can be employed.

Similar to the desktop situation, the user's calendar can be accessed for uninterruptible meetings, such as presentations, or other meetings that should not be interrupted except for notifications that are marked as immediate pass through. Similar generalizations per the calendar as described above in the desktop setting can be employed, such as utilizing information a respective manner that is provided in desktop settings or have special generalizations for the mobile or endpoint settings.

In another aspect, set time of day constraints can be provided to restrict notifications during certain times (*e.g.*, late at night and early morning, weekends). Users can specify classes of alerts they will receive to certain times. For example, all business related email and stock information will not be sent to a mobile device on weekends.

Messages sent to a cell phone or pager can be journaled by a notification manager and available when the user returns to the desktop in a notification journal view—or accesses a journal view on the mobile or endpoint device. Similar chunking of alerts can be employed for the mobile setting as for the desktop, described above.

Mobile or endpoint devices such as embedded auto personal computers (AutoPCs) and appropriately instrumented hand-held personal computers (HPCs) (*i.e.*, that have accelerometers) can be employed with the present invention. For these devices, presence information is used to infer they are active based on touch and/or acceleration, for example. A list of likely free states is created for some significant and/or distinct mobile settings (*e.g.*, a set of states each for the case of driving and for walking). For example, for driving, free states can include "just stopped at a red light or other stop and there's no conversation," or "cruising at a relatively constant velocity," for example.

Other systems can also consider different levels of attention (*e.g.*, considering speed, complexity of breaking, steering, *etc.*) For HPC's, it can be inferred (*e.g.*, Bayesian inference) with accelerometers that a user is in a car, and infer similar distinctions without direct feeds from an onboard automobile computer. For HPC's, it can be detected when devices have just been picked up, when walking or running has just ceased, or conversation has ceased, or when the unit has just been placed down to rest. For such mobile or endpoint devices, notifications can be cached locally and rendered per

likely free states. If there are no detectable two-way connections, such information can be provided in a journal such as a desktop Notification Journal as having been sent to the mobile device. It is to be appreciated that a Notification Journal can also be maintained by the endpoint device.

Users can configure the notification system so that when a user first returns to a desktop (or laptop device) after an "away state" has been detected, a single notification can be relayed, the mobile notification journal, and enable users to select particular items to view the notification that would have been observed if the user had been at the desktop. For example, users may not have a mobile device, or not have the mobile device in service, or desire to simply specify that the notification system to work in a "desktop only" modality. In this case, the following can be performed:

When the notification system notes that a user has transitioned from a "user away" to a "user present at desktop device," users are presented with a notification journal for all notifications that have gone over the max deferral time while they were away—or, per a user's preferences, foregoing the max deferral time and post all alerts to such a journal (*e.g.,* sorted in a variety of ways per user preferences, by message class, by priority, or by date, or such combinations as message class containing the highest urgency alert, sorted within class by priority or by time, *etc.*). When the user is detected to be away, notifications can additionally continue to post on the desktop (*e.g.,* in a pre-assigned area) a notification journal and continue to populate the journal (and sort by priority or by time of notification) with notifications that have gone over their max deferral time—or, alternatively foregoing the max deferral time and post substantially all alerts to such a journal. When such a journal is present, the user can be alerted with an audio cue--upon return or log in--that a journal is waiting for them. The display suppressed and rendered as an audio cue upon return and have the user take action to bring up the journal. In settings where users have been utilizing a mobile device, a journal can automatically remove journal items from the desktop journal when they are sent to the mobile device, or mark the notifications as having been transmitted to the mobile device, in order that users can sort and/or quickly scan for items they have not yet observed. Rather than posting a journal, a decision can be made to display a notification

journal, chunked alerts (per the chunking policy mentioned above), or a single alert, depending on the quantity of journaled items.

Additionally, users can be enabled to specify that the notification system delay such a "display upon return" policy, and allow users to get to work when they return (to avoid the frustration with being hit by alerts when they desire to return and get something done), and/or wait for the next "likely free" state to appear. A special "pass through" can be provided for notifications immune to such suppression. For such a functionality, additional "likely free" state to be can be defined as: "user away and returns and does not begin active work with an application or with the system." That is, it can be detected if users, upon returning to their desktop or endpoint device, begin work right away, and instead, wait until a "likely available" state is reached. If the user returns and does not begin work, this new likely free state is noted and thus causing a display of the notifications that are pending. If the user returns and is busy, the system can display notifications that have exceeded their max deferral, or, per user preference, display nothing until the next "likely free" state appears. At this time, the journal, chunked alerts, or single alerts are displayed to the user, depending on the quantity of journaled items.

Users employing a mobile or endpoint device may have the device turned off or be in a region without service. Turning on the cell phone may eventually work in a similar manner as returning to a desktop. That is, a journal view of unseen alerts may appear and users can browse and bring up respective alerts. Other aspects of the present invention can enable desktop journals to be updated when messages are reviewed on a mobile device, for example.

Turning to Fig. 3, a diagram 300 illustrates exemplary bounded deferral parameters in accordance with an aspect of the present invention. At 310, parameters on bounded deferral can be locally computed or determined based on local analysis of identity of a sender or nature of message content. At 320, these parameters can be received as part of metadata or control data from a central notification manager (*i.e.*, the bounded deferral or recommended bounded deferral can be included in a notification schema of the message). At 324, deferral parameters can be a simple function of a variable set by another user (*e.g.*, a priority flag set by message sender to indicate

urgency of message) or a function of a priority value or other parameter set by a notification manager. At 330, parameters can include bounds that are substantial —just wait for a good time, and/or erase or log if too late (over some deadline for expiration).

Referring now to Fig. 4, a system 400 illustrates various application models 410 for endpoint devices in accordance with an aspect of the present invention. The application models 410 describe some example sensing components that can be employed by endpoint devices. Sample applications 410 include use of local calendar information at 414 to guide a device to hold alerts until after a movie or show, or other meeting has ended. At 420, microphones or other sound receptors can be employed to sense when someone has stopped speaking, or other loud background noise, *e.g.*, car driving noise, car radio, *etc.*—attempt to defer alert until a period of quiet greater that *t* within a bound. At 424, motion devices such as accelerometers sense when a car has stopped, *e.g.*, at a red light or stop sign for an alert to be rendered, *e.g.*, within a bound; person has stopped walking, talking, *etc.*

In one example, applications consider the notion of ringing after bound is reached and then again as backup after quiet, *etc.* or other sensor condition. Another case waits until a person is in proximity or has touched a device before delivering a notification. Application models 410 can also consider automated deferral of phone In a simple case: Silence first few when rings quiet, while waiting for speech to stop. A more elaborate case employs an agent picking up a phone if a user is sensed to be temporarily busy and asking the user to hold on for just a bit (the bounded deferral for a call), then buzzing through when either user has stopped speaking or the bound has been reached, (*e.g.*, a minute)...coupling this, with, "please hold on for just another 30 seconds.. still trying." If this fails, directing callers to an answering machine or piping the caller through to the user. This concept can be coupled with performing phone ringing deferral only for special people or people within specially indicated groups ("People I am meeting with within an hour, *etc.*") Other sensors that may be employed by endpoint devices include Global Positioning Systems (GPS) at 430, 802.11 signal strength sensors at 434, infrared proximity sensors at 440, and touch sensors at 444.

Fig. 5 illustrates an example endpoint device 500 in accordance with an aspect of the present invention. The endpoint device 500 includes a sensing personal digital

assistant (PDA), outfitted with multiple perceptual sensors, including proximity, motion, and touch sensors. In the background at 510, accelerometer signals are displayed showing a motion fingerprint of a user walking while looking at the device 500. As highlighted in Fig. 5, small devices can be made aware of the attentional status and location of users —and transmitting local sensor information to inform a central Notification Manager, performing entirely local notification management and related services based on observations, and/or performing a combination of central and local deliberation about notification. In the latter case, the central Notification Manager makes general decisions about routing, and relies on the endpoint device to perform precision targeting of the timing and alerting modality, based on local sensing and reasoning.

As an example, with the use of bounded deferral policies, a local device commits to relaying a message that it has received before a message-specific deadline is reached; the device attempts its best to determine a suitable time for interruption within an allotted period. As noted above, the endpoint devices may embed and leverage multiple perceptual sensors on small devices, including GPS, 802.11 signal strength, accelerometers, infrared proximity detectors, and touch sensors, for example. Other aspects can exploit opportunities for developing devices, such as situation at hand, including states derived from coarse models of attention.

Figs. 6 and 7 illustrate methodologies for providing bounded deferral notifications in accordance the present invention. While, for purposes of simplicity of explanation, the methodologies are shown and described as a series of acts, it is to be understood and appreciated that the present invention is not limited by the order of acts, as some acts may, in accordance with the present invention, occur in different orders and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all illustrated acts may be required to implement a methodology in accordance with the present invention.

Referring to Fig. 6, a new notification is received at 602. At 604 the received notification is placed onto a message queue. At 606, a determination is made as to whether the received notification should be immediately passed through to the user. This

can be achieved by observing a setting such as a flag indicating whether the notification should be passed through. If the notification should be passed through, the process proceeds to 720 depicted in Fig. 7. If the notification should not be passed through, the process proceeds to 608. At 608, an initial time is associated with the notification such as a max deferral time described above. It is noted that acts 610, 612 and 616 can be executed as part of a clocked service routine or as an interrupt event, wherein these acts are periodically executed from portions of the process depicted in Figs. 6 and 7. At 610, the age of queued notifications are updated. At 612, a determination is made as to whether a notification has expired. If so, the expired notification is removed from the queue. If notifications have not expired at 612, the process returns/proceeds to the process depicted in Fig. 7.

Referring now to Fig. 7, a decision is made at 720 regarding the branch from 606 of Fig. 6. At 720, a determination is made as to whether the user is at the desktop or endpoint device. If so, the process proceeds to 724 wherein the specific notification is removed from the queue, the notification is displayed, and a notification journal is updated. If the user is not present at the desktop at 720, a determination is made at 728 whether a user mobile or endpoint device is enabled. If not, the process updates the notification journal. If the mobile device is enabled at 728, the process proceeds to 730. At 730, a determination is made as to whether a calendar indicates an uninterruptible meeting. If so, the notification journal is updated and the user is alerted after the meeting. If such a meeting is not in place at 730, the notification is transmitted to the mobile device or endpoint device and the notification journal is updated.

Referring to 740, a return is provided from the acts of 610-616 depicted in Fig. 6. At 740, a determination is made as to whether the user is present at the desktop or endpoint device. If so, a determination is made at 742 as to whether any notifications have reached the max deferral time set at 608 of Fig. 6. If so, the process proceeds to 744 and removes the specific notification from the queue and proceeds to 728 which has previously been described. At 740, if the user has just returned to the desktop or endpoint device, unseen notifications are rendered and the notification journal is updated. If the user has been at the desktop at 740, the process proceeds to 748. At 748 a determination is made as to whether any likely available states have been detected. If so, pending

notifications are rendered and the notification journal is updated. If a likely available state has not been detected at 748, the process proceeds to 750. At 750, a determination is made as to whether any notifications have reached the max deferral time set at 608 of Fig. 6. If so, the process proceeds to 724 and removes the specific notification from the queue and proceeds to display the notification and update the notification journal.

Figs. 8-14 described various aspects of notification processing and attentional modeling. As noted above, the present invention considers how systems may access and use information about a user's attention. Subtle clues about attention are often available, and a number of these clues can be taken as direct signals about the attentional status of users. For example, sensing patterns of simple gestures such as the touching and lifting of a device in different settings can relay evidence about attention that can be exploited in a number of exciting ways. Moving to higher-precision sensing, the present invention can employ the use of gaze-tracking systems, and signals about the focus of visual attention in a variety of applications. As gaze sensors grow in reliability and decrease in cost, various devices can be utilized that recognize when and how they are interrogated by the spotlight of visual attention.

Nonetheless, there often is *uncertainty* about a user's attentional focus and workload in light of observations, and about the value of alternate actions in different contexts. Thus, the present invention employs models that can be harnessed to *reason* about a user's attention and about the ideal attention-sensitive actions to take under uncertainty. Such models and reasoning can unleash new functionalities and user experiences. In some cases, models can be constructed by hand and/or learned from data Bayesian models that can be viewed as performing the task of an automated "attentional detector," working to reveal current or future attention *under uncertainty* from an ongoing stream of clues. Bayesian attentional models take as inputs sensors that provide streams of evidence about attention and provide a means for computing probability distributions over a user's attention and intentions.

Fig. 8 is a high-level decision model 800 considering a user's attentional focus and workload as a random variable, influenced by the observed states of several sensors. *Perceptual* sensors include microphones listening for ambient acoustical information or

utterances, cameras supporting visual analysis of a user's gaze or pose, accelerometers that detect patterns of motion of devices, and location sensing *via* GPS and analysis of wireless signals. However, more traditional sources of events can also offer valuable clues. These sources include a user's online calendar and considerations of the day of week and time of day. Another rich stream of evidence can be harvested by monitoring a user's interactions with software and devices. Finally, background information about the history of a user's interests and prior patterns of activities and attention can provide valuable sources of information about attention.

To build probabilistic attentional models 800 with the ability to fuse evidence from multiple sensors, the present invention leverages representations for reasoning and decision making under uncertainty. Such work has led to inferential methods and representations including Bayesian networks and influence diagrams—graphical models that extend probabilistic inference to considerations of actions under uncertainty. Algorithms can be developed which enable computing probability distributions over outcomes and expected utilities of actions from these graphical representations.

Fig. 8 displays a high-level influence diagram representing sensor fusion and decision making in the context of a user's attention under uncertainty. As portrayed in the figure, a set of variables (oval nodes) representing sensed evidence influence a random variable representing a user's attentional status which, in turn, influences the expected value of alternate actions or configurations. Intermediate cost and benefit variables can be introduced with different outcomes. Decisions (rectangular node) about ideal computer actions take into consideration the costs and benefits, given uncertainty about a user's attention. In the end, an expected utility (diamond-shaped node) is influenced by the action and the costs and benefits.

A high-level, pedagogical view can be extended by constructing richer models that contain additional intermediate variables and key interdependencies among the variables. Also, as both devices and people are immersed in time, the present invention goes beyond point-wise considerations of the states of variables, to build higher-fidelity temporal attentional models that represent changing observations and beliefs with the flow of time. This can include dynamic Bayesian networks and Hidden Markov Models for representing and reasoning about states of attention and location over time.

Fig. 9 illustrates a temporal Bayesian attentional model 900, highlighting key dependencies (dashed arcs) between variables in adjacent time slices. The model 900 displays two adjacent time slices of a temporal attentional model. Such a model provides a probability distribution over a user's workload and task that was developed for an application that provides selective filtering of messages and communications to users. In this case, the status of attention includes approximately 15 discrete states (can include more or less states).

Fig. 10 depicts a conceptual overview of a Notification Platform 1000, a cross-device messaging system that balances the costs of disruption with the value of information from multiple message sources. The system employs a probabilistic model of attention and executes ongoing decision analyses about ideal alerting, fidelity, and routing. In some cases, the Notification Platform 1000 has centered on formulating economic principles of attention-sensitive notification—and on implementing a cross-device alerting system based on these principles.

The Notification Platform 1000 modulates the flow of messages from multiple sources to devices by performing ongoing decision analyses. These analyses balance the expected value of information with the attention-sensitive costs of disruption. As highlighted in Fig. 10, the system 1000 serves as an attention-savvy layer between incoming messages and a user, taking as inputs sensors that provide information about a user's attention, location, and overall situation.

The design of the Notification Platform was informed by several earlier prototypes exploiting context-sensing for identifying a user's workload, including an automated prioritization system (See Figs. 14 and 15 below). Prioritization systems employ rules or statistical classifiers that assign values of urgency to incoming email or other messages. The classifiers are trained with sample messages, either obtained *via* explicit training or by automatically drafting data sets by observing a user's interaction with an email browser or other system. Prioritization systems may include components that also observe a user's patterns of presence at a desktop computer based on time of day, and infers the time until a user will review unread messages. The system computes an *expected cost of delayed review* for incoming sensing and calendar information, in

automated decisions about if and how to alert and transmit information to a user about email, tasks, and appointment reminders in mobile and desktop settings.

Fig. 11 illustrates an exemplary constellation of components of a Notification Platform 1100, depicting a subscription architecture. Subscribed sources and devices communicate with a Notification Manager *via* a set of standard interfaces. Sensor findings from multiple devices are considered in deliberations about information value, attention, and the best channel and alerting modality. The Notification Platform 1000 uses a decision-analytic model for cross-device alerting about information from multiple message sources. The analyses consider a user's attention and location under uncertainty, as well as the fidelity and relevance of potential communication channels. Fig. 11 displays a schematized view of the architecture of the Notification Platform 1000.

Standard interfaces and metadata schemas allow users to subscribe to different sources of information and devices to a Notification Manager. At the heart of the Notification Manager is a Bayesian attention model and decision analysis which accesses clues about attention and location from sensors *via* a module referred to as a Context Server. The Context Server accesses several states and streams of evidence, including a user's appointments from an electronic calendar (*e.g.*, Microsoft Outlook), events about device presence and activity, an analysis of ambient acoustics in the room, and a visual analysis of pose using a Bayesian head-tracking system. Key abstractions from the evidence, such as "voice trace detected," posted to a volatile store called a Context Whiteboard which is periodically updated by incoming evidence. The Context Whiteboard is typically contacted for updated information every few seconds by the Bayesian attentional model in the Notification Manager.

The Notification Manager's decision analysis weighs the expected costs and benefits of alerting a user about messages coming into the system's Universal Inbox. In computing the costs of disruption, the decision model considers the probability distribution over a user's attentional state and location in several places in its analysis, including the cost of disruption associated with different alerts for respective devices, the availability of different devices, and the likelihood that the information will reach the user when alerted in a specific manner on a device.

The ongoing expected-utility analysis is performed in accordance with a user's preferences, stored in a profile. These include assertions about the cost of disruption for each alert modality, conditioned on the user being in different attentional states. As an example, for the case of a desktop computer, the system makes available a set of display alternatives as the product of different visual displays of the alert (*e.g.*, thumbnail, full-display alert) and several auditory cues (*e.g.*, no auditory clue, soft chime, louder herald). The placement of the alert with regards to the current focus of visual attention or interaction is also considered and processed.

Fig. 12 illustrates a graphical depiction 1200 of a Notification Manager's analyses. Attention-sensitive costs of disruption and the value of information are considered, along with the losses based in decreased fidelity (narrowing funnel) and transmission reliability (spinning slotted disk) associated with the use of each alerting modality of subscribed devices. With respect to transmission reliability (transrel) of an endpoint or source device, a scale can be provided (*e.g.*, on 0-1) representing the probability p, p(transrel | context), that is the likelihood of getting through on the device given context, is a function, f(context) or f(sensed states).

Fig. 12 captures the deliberation of a Notification Platform about incoming messages. The system computes an expected value of receiving an alert as the difference between the value of alerting the user now and the value that will be obtained when the information is viewed later. Given probability distributions over a user's attention and location inferred from its sensors, Notification Platform iterates over alerting and display modalities for respective devices with an expected-utility analysis to decide if, when, and how to alert a user.

As represented with the metaphor of a narrowing funnel in Fig. 12, the system considers, for respective devices and modality, the loss in fidelity of information transmitted. In addition, the system considers the likelihood that an alert will be received, given inferred probability distributions over the attention and location of the user. This reliability of transmission is represented metaphorically in the figure as the chance that a message will make pass through a slot in a spinning disk. In the end, the attention-sensitive costs of disruption are subtracted from estimates of the value of

alerting, yielding a net value of alerting a user for each channel and alerting modality. The channel and modality with the highest expected value is selected.

Fig. 13 illustrates a view 1300 of a portion of a Notification Platform's real-time reasoning capabilities. Information from multiple sensors is posted to a Context Whiteboard and fused to infer the user's attentional status and location. Multiple notifications are sorted by net expected value, a channel, and/or an alerting modality with the highest expected utility is selected.

Fig. 13 displays several aspects of the behind-the-scenes functioning of the Notification Platform. A context palette displays current findings drawn from sensor sources. Several views onto components of the decision analysis are displayed, including inference about the time-varying attention of the user. At the current time, the user is inferred to be most likely in a state named "high-focus solo activity," which has completed recently with "low-focus solo activity," "conversation in office," and other less likely states. A Universal Inbox displays messages from several sources, including email, instant messaging, breaking news, and stock prices. Messages have also been received from *DocWatch*, a file-scouting agent subscribed to by the user that identifies documents of interest for the user.

Respective messages are annotated with the best device and alerting policy, and the associated net expected dollar value of relaying the messages with that channel and mode is indicated. As portrayed in the inbox, it is worthwhile passing on to the user two instant messages. Other alerts are "in the red," as the cost of disruption dominates the net value of information. In this case, the ideal alerting mode and channel for an instant message is determined to be a visual notification in a large format coupled with an audio herald at the user's desktop system. Users can adjust sliders to change a set of predefined defaults on costs of interruptions. Another area centers on using machine learning for building probabilistic models of attention, location, and cost of disruption from data.

Referring to Fig.14, a system 1410 illustrates a prioritization system 1412 and notification architecture in accordance with an aspect of the present invention. The prioritization system 1412 receives one or more messages or notifications 1414, generates a priority or measure of importance (*e.g.*, probability value that the message is of a high or low importance) for the associated message, and provides the one or more messages

with an associated priority value at an output 1416. As will be described in more detail below, classifiers can be constructed and trained to automatically assign measures of prioritization to the messages 1414. For example, the output 1416 can be formatted such that messages are assigned a probability that the message belongs in a category of high, medium, low or other degree category of importance. The messages can be automatically sorted in an in box of an e-mail program (not shown), for example, according to the determined category of importance. The sorting can also include directing files to system folders having defined labels of importance. This can include having folders labeled with the degree of importance such as low, medium and high, wherein messages determined of a particular importance are sorted to the associated folder. Similarly, one or more audio sounds or visual displays (*e.g.,* icon, symbol) can be adapted to alert the user that a message having a desired priority has been received (*e.g.,* three beeps for high priority message, two beeps for medium, one beep for low, red or blinking alert symbol for high priority, green and non-blinking alert symbol indicating medium priority message has been received).

According to another aspect of the present invention, a notification platform 1417 can be employed in conjunction with the prioritization system 1412 to direct prioritized messages to one or more notification sinks accessible to users. As will be described in more detail below, the notification platform 1417 can be adapted to receive the prioritized messages 1416 and make decisions regarding when, where, and how to notify the user, for example. As an example, the notification platform 1417 can determine a communications modality (*e.g.,* current notification sink 1418 of the user such as a cell phone, or Personal Digital Assistant (PDA)) and likely location and/or likely focus of attention of the user. If a high importance e-mail were received, for example, the notification platform 1417 can determine the users location/focus and direct/reformat the message to the notification sink 1418 associated with the user. If a lower priority message 1416 were received, the notification platform 1417 can be configured to leave the e-mail in the user's in-box for later review as desired, for example. As will be described in more detail below, other routing and/or alerting systems 1419 may be utilized to direct prioritized messages 1416 to users and/or other systems.

Turning now to Fig. 15, a system 1500 illustrates how a notification engine and context analyzer function together according to an aspect of the present invention. The system 1500 includes a context analyzer 1522, a notification engine 1524, one or more notification sources 1 through N, 1526, 1527, 1528, a prioritization system 1530, which can operate as a notification source, and one or more notification sinks, 1 through M, 1536, 1537, 1538, wherein N an M are integers, respectively. The sources are also referred to as event publishers, while the sinks are also referred to as event subscribers. There can be any number of sinks and sources. In general, the notification engine 1524 conveys notifications, which are also referred to as events or alerts, from the sources 1526-1528 to the sinks 1536-1538, based in part on parametric information stored in and/or accessed by the context analyzer 1522.

The context analyzer 1522 stores/analyzes information regarding variables and parameters of a user that influence notification decision-making. For example, the parameters may include contextual information, such as the user's typical locations and attentional focus or activities per the time of day and the day of the week, and additional parameters conditioned on such parameters, such as the devices users tend to have access to in different locations. Such parameters may also be functions of observations made autonomously *via* one or more sensors. For example, one or more profiles (not shown) may be selected or modified based on information about a user's location as can be provided by a global positioning system (GPS) subsystem, on information about the type of device being used and/or the pattern of usage of the device, and the last time a device of a particular type was accessed by the user. Furthermore, as is described in more detail below, automated inference may also be employed, to dynamically infer parameters or states such as location and attention. The profile parameters may be stored as a user profile that can be edited by the user. Beyond relying on sets of predefined profiles or dynamic inference, the notification architecture can enable users to specify in real-time his or her state, such as the user not being available except for important notifications for the next "x" hours, or until a given time, for example.

The parameters can also include default notification preference parameters regarding a user's preference as to being disturbed by notifications of different types in different settings, which can be used as the basis from which to make notification

decisions by the notification engine 1524, and upon which a user can initiate changes. The parameters may include default parameters as to how the user wishes to be notified in different situations (*e.g.*, such as by cell phone, by pager). The parameters can include such assessments as the costs of disruption associated with being notified by different modes in different settings. This can include contextual parameters indicating the likelihoods that the user is in different locations, the likelihoods that different devices are available, and the likelihoods of his or her attentional status at a given time, as well as notification parameters indicating how the user desires to be notified at a given time.

Information stored by the context analyzer 1522, according to one aspect of the present invention is inclusive of contextual information determined by the analyzer. The contextual information is determined by the analyzer 1522 by discerning the user's location and attentional status based on one or more contextual information sources (not shown), as is described in more detail in a later section of the description. The context analyzer 1522, for example, may be able to determine with precision the actual location of the user *via* a global positioning system (GPS) that is a part of a user's car or cell phone. The analyzer may also employ a statistical model to determine the likelihood that the user is in a given state of attention by considering background assessments and/or observations gathered through considering such information as the type of day, the time of day, the data in the user's calendar, and observations about the user's activity. The given state of attention can include whether the user is open to receiving notification, busy and not open to receiving notification, and can include other considerations such as weekdays, weekends, holidays, and/or other occasions/periods.

The sources 1526-1528, 1530 generate notifications intended for the user and/or other entity. For example, the sources 1526-1528 may include communications, such as Internet and network-based communications, and telephony communications, as well as software services. Notification sources are defined generally herein as that which generates events, which can also be referred to as notifications and alerts, intended to alert a user, or a proxy for the user, about information, services, and/or a system or world event. A notification source can also be referred to as an event source.

For example, e-mail may be generated as notifications by the prioritization system 1530 such that it is prioritized, wherein an application program or system generating the

notification assigns the e-mail with a relative priority corresponding to the likely importance or urgency of the e-mail to the user. The e-mail may also be sent without regard to the relative importance to the user. Internet-related services can include notifications including information that the user has subscribed to, such as headlines of current news every so often, and stock quotes, for example.

Notification sources 1526-1528 can themselves be push-type or pull-type sources. Push-type sources are those that automatically generate and send information without a corresponding request, such as headline news and other Internet-related services that send information automatically after being subscribed to. Pull-type sources are those that send information in response to a request, such as e-mail being received after a mail server is polled. Still other notification sources include the following:

- e-mail desktop applications such as calendar systems;
- computer systems (*e.g.,* that may alert the user with messages that information about alerts about system activity or problems);
- Internet-related services, appointment information, scheduling queries;
- changes in documents or numbers of certain kinds of documents in one or more shared folders;
- availability of new documents in response to standing or persistent queries for information; and/or,
- information sources for information about people and their presence, their change in location, their proximity (*e.g.,* let me know when I am traveling if another coworker or friend is within 10 miles of me"), or their availability (*e.g.,* let me know when Steve is available for a conversation and is near a high-speed link that can support full video teleconferencing").

The notification sinks 1536-1538 are able to provide notifications to the user. For example, such notification sinks 1536-1538 can include computers, such as desktop and/or laptop computers, handheld computers, cell phones, landline phones, pagers, automotive-based computers, as well as other systems/applications as can be appreciated. It is noted that some of the sinks 1536-1538 can convey notifications more richly than other of the sinks. For example, a desktop computer typically has speakers and a relatively large color display coupled thereto, as well as having a higher bandwidth for

receiving information when coupled to a local network or to the Internet. Thus, notifications can be conveyed by the desktop computer to the user in a relatively rich manner. Conversely, many cell phones have a smaller display that can be black and white, and receive information at a relatively lower bandwidth, for example. Correspondingly, the information associated with notifications conveyed by cell phones may generally be shorter and geared towards the phone's interface capabilities, for example. Thus, the content of a notification may differ depending on whether it is to be sent to a cell phone or a desktop computer. According to one aspect of the present invention, a notification sink can refer to that which subscribes, *via* an event subscription service, for example, to events or notifications.

The notification engine 1524 accesses the information stored and/or determined by the context analyzer, and determines which of the notifications received from the sources 1526-1528 to convey to which of the sinks 1536-1538. Furthermore, the notification engine 1524 can determine how the notification is to be conveyed, depending on which of the sinks 1536-1538 has been selected to send the information to. For example, it may be determined that notifications should be summarized before being provided to a selected sinks 1536-1538.

The invention is not limited to how the engine 1524 makes its decisions as to which of the notifications to convey to which of the notification sinks, and in what manner the notifications are conveyed. In accordance with one aspect, a decision-theoretic analysis can be utilized. For example, the notification engine 1524 can be adapted to infer important uncertainties about variables including a user's location, attention, device availability, and amount of time until the user will access the information if there were no alert. The notification engine 1524 can then make notification decisions about whether to alert a user to a notification, and if so, the nature of the summarization and the suitable device or devices to employ for relaying the notification. In general, the notification engine 1524 determines the net expected value of a notification. In doing so, it can consider the following:

- the fidelity and transmission reliability of each available notification sink;
- the attentional cost of disturbing the user;
- the novelty of the information to the user;

- the time until the user will review the information on his or her own;
- the potentially context-sensitive value of the information; and/or,
- the increasing and/or decreasing value over time of the information contained within the notification.

Inferences made about uncertainties thus may be generated as expected likelihoods of values such as the cost of disruption to the user with the use of a particular mode of a particular device given some attentional state of the user, for example. The notification engine 1524 can make decisions as to one or more of the following:

- what the user is currently attending to and doing (based on, for example, contextual information);
- where the user currently is;
- how important the information is;
- what is the cost of deferring the notification;
- how distracting would a notification be;
- what is the likelihood of getting through to the user; and,
- what is the fidelity loss associated with the use of a specific mode of a given notification sink.

Therefore, the notification engine 1524 can perform an analysis, such as a decision-theoretic analysis, of pending and active notifications, evaluates context-dependent variables provided by information sinks and sources, and infers selected uncertainties, such as the time until a user is likely to review information and the user's location and current attentional state. Furthermore, the notification engine 1524 can access information stored in a user profile by the context analyzer 1522 in lieu of or to support a personalized decision-theoretic analysis. For example, the user profile may indicate that at a given time, the user prefers to be notified *via* a pager, and only if the notification has a predetermined importance level. Such information can be utilized as a baseline from which to start a decision-theoretic analysis, or can be the manner by which the notification engine 1524 determines how and whether to notify the user. to one aspect of the present invention, the notification platform architecture 1500 can be configured as a layer that resides over an eventing or messaging infrastructure. However, the invention is not limited to any particular eventing infrastructure.

Furthermore, the architecture can be configured as a layer that resides over a flexible distributed computational infrastructure, as can be appreciated by those of ordinary skill within the art. Thus, the notification platform architecture can utilize an underlying infrastructure as a manner by which sources send notifications, alerts and events, and as a manner by which sinks such as endpoint devices receive notifications, alerts and events, for example. The present invention is not so limited, however.

With reference to Fig.16, an exemplary environment 1610 for implementing various aspects of the invention includes a computer 1612. The computer 1612 includes a processing unit 1614, a system memory 1616, and a system bus 1618. The system bus 1618 couples system components including, but not limited to, the system memory 1616 to the processing unit 1614. The processing unit 1614 can be any of various available processors. Dual microprocessors and other multiprocessor architectures also can be employed as the processing unit 1614.

The system bus 1618 can be any of several types of bus structure(s) including the memory bus or memory controller, a peripheral bus or external bus, and/or a local bus using any variety of available bus architectures including, but not limited to, 11-bit bus, Industrial Standard Architecture (ISA), Micro-Channel Architecture (MSA), Extended ISA (EISA), Intelligent Drive Electronics (IDE), VESA Local Bus (VLB), Peripheral Component Interconnect (PCI), Universal Serial Bus (USB), Advanced Graphics Port (AGP), Personal Computer Memory Card International Association bus (PCMCIA), and Small Computer Systems Interface (SCSI).

The system memory 1616 includes volatile memory 1620 and nonvolatile memory 1622. The basic input/output system (BIOS), containing the basic routines to transfer information between elements within the computer 1612, such as during start-up, is stored in nonvolatile memory 1622. By way of illustration, and not limitation, nonvolatile memory 1622 can include read only memory (ROM), programmable ROM (PROM), electrically programmable ROM (EPROM), electrically erasable ROM (EEPROM), or flash memory. Volatile memory 1620 includes random access memory (RAM), which acts as external cache memory. By way of illustration and not limitation, RAM is available in many forms such as synchronous RAM (SRAM), dynamic RAM (DRAM), synchronous DRAM (SDRAM), double data rate SDRAM (DDR SDRAM),

enhanced SDRAM (ESDRAM), Synchlink DRAM (SLDRAM), and direct Rambus RAM (DRRAM).

Computer 1612 also includes removable/non-removable, volatile/non-volatile computer storage media. Fig. 16 illustrates, for example a disk storage 1624. Disk storage 1624 includes, but is not limited to, devices like a magnetic disk drive, floppy disk drive, tape drive, Jaz drive, Zip drive, LS-100 drive, flash memory card, or memory stick. In addition, disk storage 1624 can include storage media separately or in combination with other storage media including, but not limited to, an optical disk drive such as a compact disk ROM device (CD-ROM), CD recordable drive (CD-R Drive), CD rewritable drive (CD-RW Drive) or a digital versatile disk ROM drive (DVD-ROM). To facilitate connection of the disk storage devices 1624 to the system bus 1618, a removable or non-removable interface is typically used such as interface 1626.

It is to be appreciated that Fig 16 describes software that acts as an intermediary between users and the basic computer resources described in suitable operating environment 1610. Such software includes an operating system 1628. Operating system 1628, which can be stored on disk storage 1624, acts to control and allocate resources of the computer system 1612. System applications 1630 take advantage of the management of resources by operating system 1628 through program modules 1632 and program data 1634 stored either in system memory 1616 or on disk storage 1624. It is to be appreciated that the present invention can be implemented with various operating systems or combinations of operating systems.

A user enters commands or information into the computer 1612 through input device(s) 1636. Input devices 1636 include, but are not limited to, a pointing device such as a mouse, trackball, stylus, touch pad, keyboard, microphone, joystick, game pad, satellite dish, scanner, TV tuner card, digital camera, digital video camera, web camera, and the like. These and other input devices connect to the processing unit 1614 through the system bus 1618 *via* interface port(s) 1638. Interface port(s) 1638 include, for example, a serial port, a parallel port, a game port, and a universal serial bus (USB). Output device(s) 1640 use some of the same type of ports as input device(s) 1636. Thus, for example, a USB port may be used to provide input to computer 1612, and to output information from computer 1612 to an output device 1640. Output adapter 1642 is

provided to illustrate that there are some output devices 1640 like monitors, speakers, and printers, among other output devices 1640, that require special adapters. The output adapters 1642 include, by way of illustration and not limitation, video and sound cards that provide a means of connection between the output device 1640 and the system bus 1618. It should be noted that other devices and/or systems of devices provide both input and output capabilities such as remote computer(s) 1644.

Computer 1612 can operate in a networked environment using logical connections to one or more remote computers, such as remote computer(s) 1644. The remote computer(s) 1644 can be a personal computer, a server, a router, a network PC, a workstation, a microprocessor based appliance, a peer device or other common network node and the like, and typically includes many or all of the elements described relative to computer 1612. For purposes of brevity, only a memory storage device 1646 is illustrated with remote computer(s) 1644. Remote computer(s) 1644 is logically connected to computer 1612 through a network interface 1648 and then physically connected *via* communication connection 1650. Network interface 1648 encompasses communication networks such as local-area networks (LAN) and wide-area networks (WAN). LAN technologies include Fiber Distributed Data Interface (FDDI), Copper Distributed Data Interface (CDDI), Ethernet/IEEE 1102.3, Token Ring/IEEE 1102.5 and the like. WAN technologies include, but are not limited to, point-to-point links, circuit switching networks like Integrated Services Digital Networks (ISDN) and variations thereon, packet switching networks, and Digital Subscriber Lines (DSL).

Communication connection(s) 1650 refers to the hardware/software employed to connect the network interface 1648 to the bus 1618. While communication connection 1650 is shown for illustrative clarity inside computer 1612, it can also be external to computer 1612. The hardware/software necessary for connection to the network interface 1648 includes, for exemplary purposes only, internal and external technologies such as, modems including regular telephone grade modems, cable modems and DSL modems, ISDN adapters, and Ethernet cards.

Fig. 17 is a schematic block diagram of a sample-computing environment 1700 with which the present invention can interact. The system 1700 includes one or more client(s) 1710. The client(s) 1710 can be hardware and/or software (*e.g.*, threads,

processes, computing devices). The system 1700 also includes one or more server(s) 1730. The server(s) 1730 can also be hardware and/or software (*e.g.,* threads, processes, computing devices). The servers 1730 can house threads to perform transformations by employing the present invention, for example. One possible communication between a client 1710 and a server 1730 may be in the form of a data packet adapted to be transmitted between two or more computer processes. The system 1700 includes a communication framework 1750 that can be employed to facilitate communications between the client(s) 1710 and the server(s) 1730. The client(s) 1710 are operably connected to one or more client data store(s) 1760 that can be employed to store information local to the client(s) 1710. Similarly, the server(s) 1730 are operably connected to one or more server data store(s) 1740 that can be employed to store information local to the servers 1730.

What has been described above includes examples of the present invention. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the present invention, but one of ordinary skill in the art may recognize that many further combinations and permutations of the present invention are possible. Accordingly, the present invention is intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims. Furthermore, to the extent that the term "includes" is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term "comprising" as "comprising" is interpreted when employed as a transitional word in a claim.